



Data Collection, Security, Storage & Management Services Bundles - Beta Release

D5.2

December 2021

Deliverable

PROJECT ACRONYM	GRANT AGREEMENT #	PROJECT TITLE
TWINERGY	957736	Intelligent interconnection of prosumers in positive energy communities with twins of things for digital energy markets

DELIVERABLE REFERENCE NUMBER AND TITLE

D5.2 Data Collection, Security, Storage & Management Services Bundles – Beta Release

Revision: 1

AUTHORS

Alexandros Tsitsanis	Evgenia Gkilpathi
Suite5	Suite5



Funded by the Horizon 2020 programme of the European Union
Grant Agreement No 957736

DISSEMINATION LEVEL

✓ P Public

C Confidential, only for members of the consortium and the Commission Services

Version History

REVISION	DATE	AUTHORS	ORGANISATION	DESCRIPTION
v1.0	23/12/2021	Alexandros Tsitsanis, Evgenia Gkilpathi	Suite5	1st draft of Deliverable D5.2 "Data Collection, Security, Storage, and Management Services Bundles"
	24/12/2021	Konstantinos Kotsalos, Apostolos Kapetanios	ED	Minor Corrections/Changes and Suggestions
	27/12/2021	Ana Isabel Martínez García, Moisés Antón García	ETRA	Suggestions for additional references/Minor Changes
	29/12/2021	Fynn Christian Bollhöfer, Axel Balke	TH-OWL	Minor text changes and Suggestions
V1.1 Consolidated Version – Ready for Submission	– 30/12/2021	Alexandros Tsitsanis, Evgenia Gkilpathi	Suite5	Ready for submission

Statement of Originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Legal Disclaimer

The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. The above referenced authors will have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law. The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the European Union. Neither the INEA nor the European Commission is responsible for any use that may be made of the information contained therein.

© 2021 by TwinERGY Consortium

Executive Summary

Deliverable D5.2 "Data Collection, Security, Storage, and Management Services Bundles" summarizes three (3) tasks, namely Task 5.2 "Data Management Platform Backbone Infrastructure," Task 5.3 "Core Data Ingestion, Curation, and Management Services," and Task 5.4 "Data security, encryption, and privacy mechanisms" of WP5 "Data Collection and Communication Platform", as depicted in the TwinERGY DoA[1]. The current deliverable provides the beta release of the respective services bundles. The data availability in the pilot sites, the data requirements of the TwinERGY modules, and the nine (9) use cases of the TwinERGY project were analysed, as explained in the deliverable D5.3 "TwinERGY Integrated Data Management Platform – Alpha, Mock-ups Release," in order to define the functional requirements for the development of the TwinERGY CDMP and the different services that will be offered by the Platform. Each service introduces its own set of functionalities.

This deliverable dives into the development details of the services, covering the following aspects:

- The current status of various functionalities provided from each service, as well as prospective functionalities to be considered in future TwinERGY Core Data Management Platform releases.
- The frameworks and libraries used to implement each service, as well as their versions.
- The interface specifications that were created, in order to document the technical details of each service's APIs.

The current deliverable is the beta version of TwinERGY Core Data Management Platform's services on M14. The delivered services were integrated at the mock-up level (in the alpha release detailed in TwinERGY Deliverable D5.3 ("TwinERGY Integrated Data Management Platform – Alpha, Mock-ups Release" [3]) setting the basis for the development activities of the TwinERGY Integrated Data Management Platform Beta Release (D5.4) on M16 (February 2022).

Index

1. Introduction	2
1.1 Purpose of this deliverable	2
1.2 Scope of this deliverable	2
1.3 Structure of the document	3
1.4 Abbreviation List	3
2. TwinERGY Core Data Management Platform and Services Overview	5
2.1 TwinERGY Core Data Management Platform	5
2.2 Services overview	5
2.2.1 Data Collection Service	5
2.2.1.1 State of Development	6
2.2.1.2 Technological Infrastructure	7
2.2.1.3 Interface Specification	8
2.2.2 Data Security Service	12
2.2.2.1 State of Development	13
2.2.2.2 Technological Infrastructure	13
2.2.2.3 Interface specification	13
2.2.3 Data Storage Service	16
2.2.3.1 State of Development	16
2.2.3.2 Technological Infrastructure	16
2.2.3.3 Interface Specification	17
2.2.4 Platform Management Service	17
2.2.4.1 State of Development	17
2.2.4.2 Technological Infrastructure	18
2.2.4.3 Interface Specification	18
2.3 Services Installation and Access	19
2.3.1 Services Installation	20
2.3.2 Licensing and access	20
3. Conclusions	21
REFERENCES	22

List of Figures

Figure 1 The TwinERGY Core Data Management Platform conceptual architecture 5

1. Introduction

1.1 Purpose of this deliverable

The project's backbone is an "open," modular, and interoperable big data management platform that will enable open standards-based data collection and management communication across the project's energy value chain, as depicted in the TwinERGY DoA [2]. The TwinERGY Data Management Platform will follow existing open energy standards (e.g., OpenADR, USEF, etc.), while also incorporating a homogenized Common Information Model that ensures semantic interoperability of the digitalized energy assets used in the pilot sites, as well as seamless integration, communication, and operation on top of any Energy Management System and Smart Home systems and devices. The TwinERGY Data Management Platform will be provided with proper data security, privacy, authentication, and authorization techniques to ensure end-user data confidentiality and non-repudiation of DER assets.

In order to define the functional requirements for the development of the TwinERGY CDMP and the different services that will be offered by the platform, a thorough analysis of the data availability in the pilot sites, the data requirements of the TwinERGY modules, and the way that the TwinERGY use cases are covered was conducted, as documented in deliverable D5.3 "TwinERGY Integrated Data Management Platform – Alpha, Mock-ups Release" [3], which provided the conceptual architecture of the TwinERGY Core Data Management Platform, accompanied with relevant mock-ups.

The Data Collection, Data Security, Data Storage, and Platform Management services, as well as the various functionalities introduced per service, are properly explained in this deliverable. D5.4 "TwinERGY Integrated Platform– Beta Release" will cover the TwinERGY CDMP's User Interface implementation of the various services.

1.2 Scope of this deliverable

The major scope of D5.2 is to provide a comprehensive overview of the TwinERGY Core Data Management Platform Services beta version. D5.2 goes through the TwinERGY CDMP's services in greater detail, as well as their functionalities. Data Collection, Data Security, Data Storage, and Platform Management services, are all described in detail by:

- Depicting the implemented functionalities for each service, as well as the functionalities in development and those that will be considered for future TwinERGY Data Management Platform releases if they are deemed required.
- Describing the libraries and frameworks used to implement each service.
- Documenting the development of relevant APIs that will make communication and information sharing across services easier.

1.3 Structure of the document

The structure of the document is as follows:

Section 2 provides a brief description of the TwinERGY Core Data Management Platform conceptual architecture, the services and their functionalities, setting the basis for the next section. It also provides an overview of the platform services and shows how the functionalities of each service are being developed, as well as the technological infrastructure that was used to establish each service and the specifications of the various interfaces that have been created, together with the appropriate licensing and access methods.

Deliverable D5.2 "Data Collection, Security, Storage, and Management Services Bundles" is concluded with Section 3, which provides a brief synopsis of what has been covered in the various sections of the deliverable.

1.4 Abbreviation List

Acronym	Full Name
API	Application Programming Interface
CDMP	Core Data Management Platform
CIM	Common Information Model
CSV	Comma-separated values
D	Deliverable
DoA	Description of Action
DER	Distributed Energy Resources
DMP	Data Management Platform
GDPR	General Data Protection Regulation
JSON	JavaScript Object Notation
M	Month
UI	User Interface
WP	Work Package
XML	Extensible Mark-up Language

2. TwinERGY Core Data Management Platform and Services Overview

2.1 TwinERGY Core Data Management Platform

Data Collection, Data Security, Data Storage, and Platform Management are all part of the TwinERGY Core Data Management Platform, as depicted in Figure 1. The platform's services and their respective functionalities will be detailed and further examined in the sections that follow throughout this deliverable.

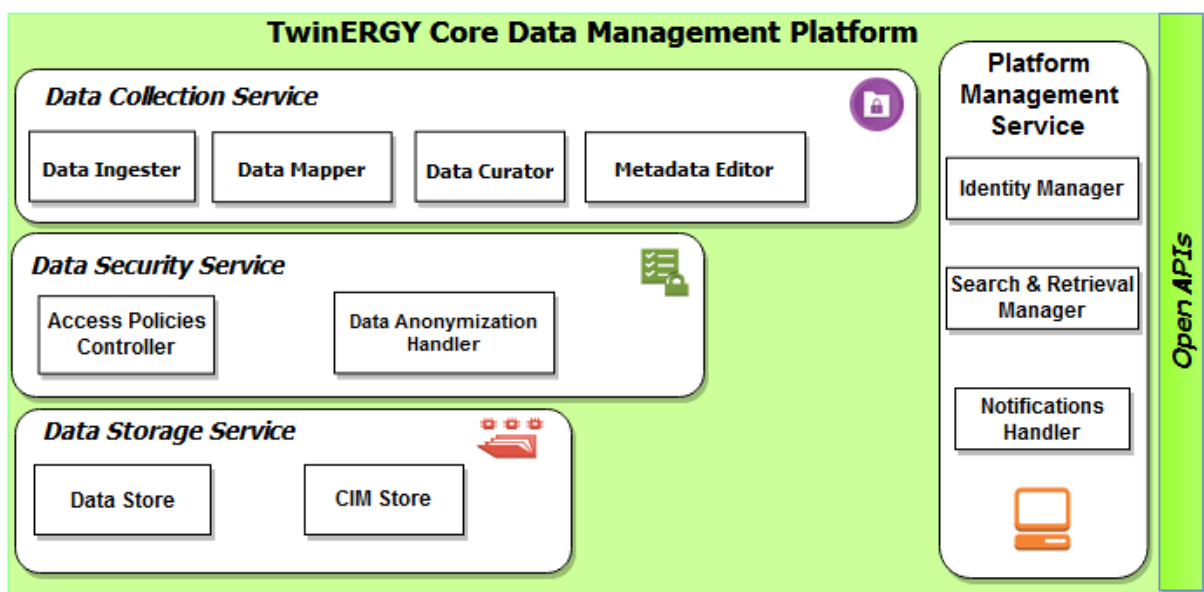


Figure 1 The TwinERGY Core Data Management Platform conceptual architecture

The above architecture of the CDMP was introduced, in order to address the requirements that were identified, after the completion of the activities and evaluation of the findings and feedback obtained from the TwinERGY Uses Cases elaboration, the TwinERGY pilot sites data availability activities and the identification of the various data requirements of the different modules to be developed in the context of the TwinERGY project.

In the following sections, a detailed overview of the services and of their functionalities is provided, along with the state of development of each service's functionalities and the interfaces specifications created to describe the technical details of each service's APIs.

2.2 Services overview

2.2.1 Data Collection Service

The Data Collection Service introduces a variety of functionalities that are relevant to the data collection process, including: i) Data Ingestor, which introduces proper

procedures and methods for ingesting data into the CDMP; ii) Data Mapper, which allows ingested datasets to be aligned with the CIM; iii) Data Curator, which ensures the datasets quality; and iv) Metadata Editor, which enables the users of the platform to define specific aspects of the ingested dataset, in order to provide a detailed profile of the given dataset.

The appropriate handling of data management procedures enables data collection setup, allowing users to specify the data ingestion method, according to which the data is uploaded to the TwinERGY CDMP. The platform integrates data from a variety of sources in a straightforward manner. Users uploading data will use the platform to clarify the way and the type of data they own and are ready to share, as well as to define the parameters of the data ingestion process, outline their needs and preferences on the way that the data will be inserted in the platform (users APIs, or file uploading), and perform proper mapping actions on all fields of the data, all of which will be based on the TwinERGY Common Information Model.

The Data Mapper functionality determines how the fields of a dataset are aligned with the concepts of the TwinERGY CIM. When the CIM and the dataset are matched, a consistent domain-specific understanding of the data is established, making it easier to use for data-related services in the CDMP. To guarantee adherence with the CIM, the application of appropriate mapping and transformation routines on datasets allows for the changing of field names and the conversion of measurement units.

Additionally, during the curation configuration process, the Data Curator functionality allows the user to specify the restrictions and limitations that the data ingested into the platform may have, as well as the precise activities undertaken if any of these constraints are not satisfied. This configuration refers to a set of curation routines that are performed by carrying out the required data curation processes.

Any result of data ingestion is saved as a dataset in the TwinERGY Core Data Management Platform and with the Metadata Editor functionality, the users of the platform have the capability to specify a comprehensive dataset profile, by determining the dataset title, description and additional metadata in alignment with the DCAT [4] recommendations.

2.2.1.1 State of Development

The Data Collection Service release, as specified in this deliverable, introduces a range of functionalities that are described below.

The functionalities that are part of the beta release of the Data Collection Services can be depicted as follows:

- Data ingestion process configuration: describes in detail the needs and preferences on the way that the data will be ingested into the TwinERGY Core Data Management Platform (file uploading, API, streaming data).
- Ingestion of data from files: allows data to be retrieved from files in popular file formats (e.g., csv).
- API Data Ingestion: This functionality will allow data to be retrieved from both pilot system APIs and Open Data APIs (e.g., weather data, other local sources).

- Reliable and safe data upload: Data is uploaded to the TwinERGY Core Data Management Platform using reliable and secure processes.
- Mapping activities on ingested data: This functionality ensures that the data values are matched to the CIM provisions (e.g. data types, measurement units) by performing the necessary mapping actions on the ingested data.
- Dataset & metadata creation: Any outcome of a data ingestion is recorded as a dataset in the Data Storage Service of the TwinERGY Core Data Management Platform with this functionality. Users of the platform have the ability to give a dataset a title and associated metadata, as well as describe what is contained inside it.

2.2.1.2 Technological Infrastructure

The back - end of the Data Collection Service is developed with **Python 3.10**¹, using also a range of open-source technologies, namely **Django 4.0**², **Pandas 1.3.0**³ and **Docker 20.10.11**⁴.

Python is a high-level, general-purpose programming language, which emphasises code readability with the use of indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects⁵.

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support⁶.

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series⁷.

Docker is an open source containerization platform. It enables developers to package applications into containers—standardized executable components combining application source code with the operating system (OS) libraries and dependencies required to run that code in any environment⁸.

The front- end of the Data Collection Service is developed with **Vue.js V2**⁹.

¹ <https://www.python.org/downloads/release/python-3100/>

² <https://www.djangoproject.com/weblog/2021/dec/07/django-40-released/>

³ <https://pandas.pydata.org/docs/whatsnew/index.html>

⁴ <https://docs.docker.com/engine/release-notes/>

⁵ <https://analyticsindiamag.com/python-3-10-and-its-new-features/>

⁶ <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction>

⁷ [https://en.wikipedia.org/wiki/Pandas_\(software\)](https://en.wikipedia.org/wiki/Pandas_(software))

⁸ <https://www.ibm.com/in-en/cloud/learn/docker>

⁹ <https://vuejs.org/v2/guide/>

Vue is a progressive framework for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only and is easy to pick up and integrate with other libraries or existing projects.

2.2.1.3 Interface Specification

Name	Start a new data ingestion job		
Description	Initiates the data ingestion process		
Reference Code	DCS01		
Service	Data Collection		
Type of interface	REST		
Inputs	Methods of the interface	Parameters of the method	Return Values of the method
	POST	N/A	200 OK 201 Created 401 Unauthorized 403 Forbidden 404 Not Found
Endpoint URI	/twinergy-api/data-collection/		
Interaction with Services	Data Collection Service		

Name	Retrieve a data ingestion job		
Description	Initiate the data ingestion retrieval process		
Reference Code	DCS02		
Service	Data Collection		
Type of interface	REST		
Inputs	Methods of the interface	Parameters of the method	Return Values of the method
	GET	id: The identifier of the data collection	id (string) title (string) user (string) configuration (json) created (datetime) status (string)
Endpoint URI	/twinergy-api/data-collection/{id}		
Interaction with Services	Data Collection Service		

Name	Delete a data ingestion job		
Description	Deletes a data ingestion job		
Reference Code	DCS03		
Service	Data Collection		

Type of interface	REST		
Inputs	Methods of the interface	Parameters of the method	Return Values of the method
	DELETE	id: The identifier of the data collection	200 OK 401 Unauthorized 403 Forbidden 404 Not Found
Endpoint URI	/twinergy-api/data-collection/{id}		
Interaction with components	Data Collection Service		

Name	Ingester Configuration		
Description	Creates the data ingester settings		
Reference Code	DCS04		
Service	Data Collection		
Type of interface	REST		
Inputs	Methods of the interface	Parameters of the method	Return Values of the method
	POST	id: The identifier of the data ingester configuration	200 OK 201 Created 401 Unauthorized 403 Forbidden 404 Not Found
Endpoint URI	/twinergy-api/data-collection/data-ingester-config/{id}		
Interaction with Services	Data Collection Service		

Name	Ingester Retrieval		
Description	Retrieves the data ingester settings		
Reference Code	DCS05		
Service	Data Collection		
Type of interface	REST		
Inputs	Methods of the interface	Parameters of the method	Return Values of the method
	GET	id: The identifier of the data ingester retrieval	id (string) step-id (string) title (string) configuration (json) created (datetime) status (string)
Endpoint URI	/twinergy-api/data-collection/data-ingester-config/{id}		
Interaction with Services	Data Collection Service		

Name	Mapper Configuration		
Description	Creates the data mapper settings		
Reference Code	DCS06		
Service	Data Collection		
Type of interface	REST		
Inputs	Methods of the interface	Parameters of the method	Return Values of the method
	POST	id: The identifier of the data mapper configuration	200 OK 201 Created 401 Unauthorized 403 Forbidden 404 Not Found
Endpoint URI	/twenergy-api/data-collection/ data-mapper-config /{id}		
Interaction with Services	Data Collection Service		

Name	Mapper Retrieval		
Description	Retrieves the data mapper settings		
Reference Code	DCS07		
Service	Data Collection		
Type of interface	REST		
Inputs	Methods of the interface	Parameters of the method	Return Values of the method
	GET	id: The identifier of the data mapper retrieval	id (string) step-id (string) title (string) configuration (json) created (datetime) status (string)
Endpoint URI	/twenergy-api/data-collection/data-mapper-config/{id}		
Interaction with Services	Data Collection Service		

Name	Curator Configuration		
Description	Creates the data curator settings		
Reference Code	DCS08		
Service	Data Collection		
Type of interface	REST		
Inputs	Methods of the interface	Parameters of the method	Return Values of the method
	POST	id: The identifier of the data curator	200 OK 201 Created 401 Unauthorized 403 Forbidden 404 Not Found

Endpoint URI	/twinergy-api/data-collection/data-curator-config/{id}
Interaction with Services	Data Collection Service

Name	Curator Retrieval		
Description	Retrieves the data curator settings		
Reference Code	DCS09		
Service	Data Collection		
Type of interface	REST		
Inputs	Methods of the interface	Parameters of the method	Return Values of the method
	GET	id: The identifier of the data curator retrieval	id (string) step-id (string) title (string) configuration (json) created (datetime) status (string)
Endpoint URI	/twinergy-api/data-collection/data-curator-config/{id}		
Interaction with Services	Data Collection Service		

Name	Metadata Editor Configuration		
Description	Creates the metadata editor settings		
Reference Code	DCS10		
Service	Data Collection		
Type of interface	REST		
Inputs	Methods of the interface	Parameters of the method	Return Values of the method
	POST	id: The identifier of the metadata editor configuration	200 OK 201 Created 401 Unauthorized 403 Forbidden 404 Not Found
Endpoint URI	/twinergy-api/data-collection/metadataeditor-config/{id}		
Interaction with Services	Data Collection Service		

Name	Metadata Editor Retrieval		
Description	Retrieves the metadata editor settings		
Reference Code	DCS11		
Service	Data Collection		
Type of interface	REST		
Inputs	Methods of the interface	Parameters of the method	Return Values of the method

	GET	id: The identifier of the metadata editor retrieval	id (string) step-id (string) title (string) configuration (json) created (datetime) status (string)
Endpoint URI	/twinergy-api/data-collection/metadataeditor-config/{id}		
Interaction with Services	Data Collection Service		

2.2.2 Data Security Service

The TwinERGY CDMP users' data security and privacy concerns about data that will be ingested and handled in the TwinERGY Core Data Management Platform are addressed by the Data Security Service. As a result, it includes a number of functions and functionalities that may be easily configured in the TwinERGY Platform, such as the Access Policies Controller and the Data Anonymization Handler.

The Access Policies Controller functionality enhances user confidence in the entire TwinERGY Core Data Management Platform by allowing users to create access policies that flexibly and clearly regulate access requests to data within the TwinERGY CDMP. The access policy controller effectively handles policy enforcement at run-time so that only authorized platform users can view and access the data. This functionality allows platform users to define and configure access policy routines, allowing these routines to be based on data fields and requestor properties. Routines are saved and users uploading data can easily modify them. After the access policy is defined and configured, the policy is applied when access to the data contained in the CDMP is requested.

As for the Anonymization handler functionality, it ensures the protection of the data from inadvertent disclosure of personal and company information. Before the data becomes available on the TwinERGY platform, anonymization actions are performed and reflected in the defined part of the data (i.e., fields). The users of the CDMP will be provided with the appropriate interventions on the data in case they need to anonymize fields in their data that may contain personally identifiable information.

2.2.2.1 State of Development

The Data Security Service release, as specified in this deliverable, introduces a range of functionalities that are described below.

The functionalities that are part of the beta release of the Data Security Services can be depicted as follows:

- Backend access policy management and handling: Policies can be used to determine when access is permitted.

- Backend enforcement of data access policies: The access policies that have been set are applied when a request is made to access data that has been ingested into the TwinERGY CDMP.

2.2.2.2 Technological Infrastructure

The back - end of the Data Security Service is developed with **Python 3.10**, using also a range of open-source technologies, namely **Django 4.0**, and **Pandas 1.3.0**, as documented in 2.2.1.2.

The access policy management library of the Data Security Service is **CASL v4¹⁰**. **CASL** is an isomorphic authorization JavaScript library which restricts what resources a given client is allowed to access. It's designed to be incrementally adoptable and can easily scale between a simple claim based and fully featured subject and attribute based authorization. It makes it easy to manage and share permissions across UI components, API services, and database queries.

2.2.2.3 Interface specification

Name	Access Policy Creation		
Description	Creates an access policy for a dataset		
Reference Code	DSS01		
Service	Data Security		
Type of interface	REST		
Inputs	<u>Methods of the interface</u>	<u>Parameters of the method</u>	<u>Return Values of the method</u>
	POST	id: The identifier of the dataset	200 OK 201 Created 401 Unauthorized 403 Forbidden 404 Not Found
Endpoint URI	/twinergy-api/dataset/{id}/policy		
Interaction with Services	Data Security Service		

Name	Access Policy Retrieval		
Description	Retrieves all access policies for a dataset		
Reference Code	DSS02		
Service	Data Security		
Type of interface	REST		
Inputs	<u>Methods of the interface</u>	<u>Parameters of the method</u>	<u>Return Values of the method</u>

¹⁰ <https://casl.js.org/v4/en/guide/intro>

	GET	id: The identifier of the dataset	id (string) step-id (string) title (string) policies (json) created (datetime) status (string)
Endpoint URI	/twinergy-api/dataset/{id}/policy		
Interaction with Services	Data Security Service		

Name	Access Policy Update		
Description	Updates an access policy for a dataset		
Reference Code	DSS03		
Service	Data Security		
Type of interface	REST		
Inputs	<u>Methods of the interface</u>	<u>Parameters of the method</u>	<u>Return Values of the method</u>
	POST	id: The identifier of the dataset policy_id: The identifier of the access policy	200 OK 201 Created 401 Unauthorized 403 Forbidden 404 Not Found
Endpoint URI	/twinergy-api/dataset/{id}/policy/{id}		
Interaction with Services	Data Security Service		

Name	Access Policy Deletion		
Description	Deletes an access policy for a dataset		
Reference Code	DSS04		
Service	Data Security		
Type of interface	REST		
Inputs	<u>Methods of the interface</u>	<u>Parameters of the method</u>	<u>Return Values of the method</u>
	DELETE	id: The identifier of the dataset policy_id: The identifier of the access policy	200 OK 401 Unauthorized 403 Forbidden 404 Not Found
Endpoint URI	/twinergy-api/dataset/{id}/policy/{id}		
Interaction with Services	Data Security Service		

Name	Anonymization Handler Configuration		
Description	Creates the data anonymization handler settings		
Reference Code	DSS05		
Service	Data Security		
Type of interface	REST		
Inputs	<u>Methods of the interface</u>	<u>Parameters of the method</u>	<u>Return Values of the method</u>
	POST	id: The identifier of the data anonymization handler configuration	200 OK 201 Created 401 Unauthorized 403 Forbidden 404 Not Found
Endpoint URI	/twinergy-api/data-collection/data-anonymization-handler-config/{id}		
Interaction with Services	Data Security Service		

Name	Anonymization Handler Retrieval		
Description	Retrieves the data anonymization handler settings		
Reference Code	DSS06		
Service	Data Security		
Type of interface	REST		
Inputs	<u>Methods of the interface</u>	<u>Parameters of the method</u>	<u>Return Values of the method</u>
	GET	id: The identifier of the data anonymization handler retrieval	id (string) step-id (string) title (string) configuration (json) created (datetime) status (string)
Endpoint URI	/twinergy-api/data-collection/data-anonymization-handler-config/{id}		
Interaction with Services	Data Security Service		

2.2.3 Data Storage Service

This service meets the demand for dependable data storage and indexing by providing resilient measures and different indexing methodologies. Depending on the type of information to be stored in the TwinERGY Platform and how it will be retrieved, different storage solutions are utilised to meet various needs. The Data Storage Service is in charge of securely storing a wide range of data, as well as the metadata that goes with it, in a reliable and secure manner.

2.2.3.1 State of Development

The Data Storage Service release, as specified in this deliverable, introduces a range of functionalities that are described below.

The functionalities that are part of the beta release of the Data Storage Services can be depicted as follows:

- Data Store: This functionality entails storing various types of data (e.g., datasets) together with their associated metadata, making it accessible to all TwinERGY Core Data Management Platform services.
- CIM Store: Storage of the TwinERGY Common Information Model, as well as all necessary concepts and fields.

2.2.3.2 Technological Infrastructure

The back - end of the Data Storage Service is developed with the utilization of **PostgreSQL 14**¹¹ and **ElasticSearch 7.14**¹².

PostgreSQL is a powerful, open source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance¹³.

Elasticsearch is a distributed, free and open search and analytics engine for all types of data, including textual, numerical, geospatial, structured, and unstructured. Known for its simple REST APIs, distributed nature, speed, and scalability, Elasticsearch is the central component of the Elastic Stack, a set of free and open tools for data ingestion, enrichment, storage, analysis, and visualization¹⁴.

2.2.3.3 Interface Specification

There are no external APIs that are exposed by the Data Storage Service to accompany its initial version except for the APIs that have been presented in sections 2.2.1.3, 2.2.2.3, 2.2.4.3.

2.2.4 Platform Management Service

This service establishes the tools and processes for users to register on the platform in a safe and reliable manner. Through appropriate authentication and security measures, users are given access to the data that they are authorized to use. When data ingestion takes place in the platform, users will be alerted via platform notifications based on their preferences. The service has three main functionalities: the (i) Identity Manager, the (ii) Search & Retrieval Manager, and the (iii) Notifications Handler.

Summarized, with the development of the Platform Management Service:

¹¹ <https://www.postgresql.org/about/news/postgresql-14-released-2318/>

¹² <https://www.elastic.co/blog/whats-new-elasticsearch-7-14-0>

¹³ <https://www.postgresql.org/>

¹⁴ <https://www.elastic.co/what-is/elasticsearch>

- Users may register, authorize, and authenticate themselves using the TwinERGY CDMP's identity management functionality, which is interconnected with security-related processes.
- Users will be able to search for and find data that may be proven valuable, as well as determine and define which of these available data are relevant to their needs, using the Search & Retrieval functionality.
- The Notifications handling functionality is in charge of detecting events related to the status and progress of a data ingestion in order to provide the TwinERGY Core Data Management Platform's users with appropriate material.

2.2.4.1 State of Development

The Platform Management Service release, as specified in this deliverable, introduces a range of functionalities that are described below.

The functionalities that are part of the beta release of the Platform Management Services can be depicted as follows:

- **Identity Management:** This functionality simulates the functions of an identity provider, such as establishing and managing identifying information for those who are permitted to use the TwinERGY CDMP.

2.2.4.2 Technological Infrastructure

The back-end of the Platform Governance Module is developed with Python 3.10, using also a range of open-source technologies, namely Django 4.0, as documented in 2.2.1.2.

Websockets 10.1¹⁵ and **Kubernetes 1.22.2**¹⁶ are utilized for server-side notifications implementation and for the system deployment respectively.

Websockets is a library for building WebSocket servers and clients in Python with a focus on correctness, simplicity, robustness, and performance.

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available¹⁷.

2.2.4.3 Interface Specification

Name	User Retrieval
Description	Retrieves the specifics for a user
Reference Code	PMS01
Service	Platform Management
Type of interface	REST

¹⁵ <https://websockets.readthedocs.io/en/stable/>

¹⁶ <https://kubernetes.io/blog/2021/08/04/kubernetes-1-22-release-announcement/>

¹⁷ <https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>

Inputs	<u>Methods of the interface</u>	<u>Parameters of the method</u>	<u>Return Values of the method</u>
		GET	id: The identifier of the user
Endpoint URI	/twinergy-api/user/{id}		
Interaction with Services	Platform Management Service		

Name	User Update		
Description	Updates the specifics for a user		
Reference Code	PMS02		
Service	Platform Management		
Type of interface	REST		
Inputs	<u>Methods of the interface</u>	<u>Parameters of the method</u>	<u>Return Values of the method</u>
	POST	id: The identifier of the user	200 OK 201 Created 401 Unauthorized 403 Forbidden 404 Not Found
Endpoint URI	/twinergy-api/user/{id}		
Interaction with Services	Platform Management Service		

Name	User Deletion		
Description	Deletes a user		
Reference Code	PMS03		
Service	Platform Management		
Type of interface	REST		
Inputs	<u>Methods of the interface</u>	<u>Parameters of the method</u>	<u>Return Values of the method</u>
	DELETE	id: The identifier of the user	200 OK 401 Unauthorized 403 Forbidden 404 Not Found
Endpoint URI	/twinergy-api/user/{id}		
Interaction with Services	Platform Management Service		

Name	User Password Update		
Description	Updates the password of a user		

Reference Code	PMS04		
Service	Platform Management		
Type of interface	REST		
Inputs	<u>Methods of the interface</u>	<u>Parameters of the method</u>	<u>Return Values of the method</u>
	PUT	id: The identifier of the user	200 OK 401 Unauthorized 403 Forbidden 404 Not Found
Endpoint URI	/twinergy-api/user/{id}/password		
Interaction with Services	Platform Management Service		

2.3 Services Installation and Access

2.3.1 Services Installation

A private repository contains the Data Collection, Data Security, Data Storage, and Platform Management Services. Because all elements are already packed as Docker containers, the relevant private code repository includes explicit instructions for the service's deployment.

2.3.2 Licensing and access

Only closed-source code has been used in the development of the Data Collection, Data Security, Data Storage, and Platform Management services. The versions of the different services that are to be deployed will be accessed through the TwinERGY project's Core Data Management Platform (as documented in D5.3 for the alpha release and in the upcoming D5.4 for the beta release).

3. Conclusions

The initial version of the TwinERGY Core Data Management Platform Services was the objective of deliverable D5.3, "Data Collection, Security, Storage, and Management Services Bundles." It also included a brief report detailing the implementation progress of each service's functionalities, as well as the technological infrastructure used, and the interface specifications established. The deliverable outlined the initial version of the four (4) TwinERGY CDMP services developed, namely the Data Collection, Data Security, Data Storage, and Platform Management services, providing additional details for each one of the services including:

- The current state of development of each developed functionality, for each service
- The technological infrastructure of each service, including the version used.
- The interface specifications that were created.

The current deliverable is the first version of the services released on M14 in accordance with the TwinERGY DoA, providing back-end implementation details and laying the groundwork for the upcoming beta version of the TwinERGY Integrated Platform that will be provided on M16 (D5.4), which will include the matching User Interface implementation. It is crucial to mention, that the TwinERGY Platform's services development is a live operation and will be continued until M32. Any adjustments and improvements made as the project progresses, as well as any new requirements that arise, will be incorporated into future editions of the TwinERGY Core Data Management Platform.

REFERENCES

[1] DoA Part A, TwinERGY Consortium Agreement No 957736

[2] DoA Part B, TwinERGY Consortium Agreement No 957736

[3] TwinERGY Consortium. (2020). TwinERGY D5.3 "TwinERGY Integrated Data Management Platform – Alpha, Mockups Release"

[4] Data Catalog Vocabulary (DCAT) - Version 2 (w3.org)